

Pertemuan 1

CPMK : Mahasiswa mampu menguasai konsep dasar dari interaksi manusia dan komputer, dan mampu menyebutkan siapa saja yang terlibat dalam IMK

Sub-CPMK : Mahasiswa mampu mendeskripsikan konsep dasar interaksi manusia dan komputer.

PENDAHULUAN / DEFINISI

- Bidang ilmu interaksi manusia dan komputer adalah ilmu yang mempelajari tentang bagaimana mendesain, mengevaluasi, dan mengimplementasikan sistem komputer yang interaktif sehingga dapat digunakan oleh manusia dengan mudah.
- Pengertian Interaksi = komunikasi 2 arah antara manusia (user) dan sistem komputer.
Interaksi menjadi maksimal apabila kedua belah pihak mampu memberikan stimulan dan respon (aksi & reaksi) yang saling mendukung, jika salah satu tidak bisa, maka interaksi akan mengalami hambatan atau bahkan menuju pembiasan tujuan.
- Definisi interaksi manusia dan komputer = sebuah hubungan antara manusia dan komputer yang mempunyai karakteristik tertentu untuk mencapai suatu tujuan tertentu dengan menjalankan sebuah sistem yang bertopengkan sebuah antarmuka (interface).
- Prinsip kerja komputer = input → proses → output Kepada komputer diberikan data yang umumnya berupa deretan angka dan huruf. Kemudian diolah didalam komputer yang menjadi keluaran sesuai dengan kebutuhan dan keinginan manusia. Tanpa disadari kita (manusia/user) telah berinteraksi atau berdialog dengan sebuah benda (layar monitor), yaitu dalam bentuk menekan tombol berupa tombol angka dan huruf yang ada pada keyboard atau melakukan satu sentuhan kecil pada mouse. Yang kemudian hasil inputan ini akan berubah bentuk menjadi informasi atau data yang seperti diharapkan manusia dengan

tertampilnya informasi baru tersebut pada layar monitor atau bahkan mesin pencetak (printer)

- Manusia pada umumnya tidak pernah tahu apa yang terjadi pada saat data dimasukkan ke dalam kotak cpu melalui keyboard. Manusia (user) selalu terfokus pada monitor/printer sebagai keluaran.
- Manusia jarang sekali menyadari proses interaksi dengan komputer. Manusia baru menyadari proses interaksi tersebut saat menemukan masalah dan tidak menemukan solusi pemecahannya. Biasanya manusia menyalahkan antarmuka yang kurang inovatif, kurang menarik, kurang komunikatif.
- Interaksi bisa dikatakan dialog antara user dengan komputer.

Model atau jenis interaksi, antara lain :

1. Command line interface (perintah baris tunggal)

contoh : unix, linux, dos

2. Menu (menu datar dan menu tarik)

contoh : hampir semua software menggunakan menu

3. Natural language (bahasa alami)

contoh : bahasa pemrograman terstruktur (belum objek)

4. Question/answer and query dialogue

contoh : mysql, dbase interaktif, dll

5. Form-fills and spreadsheets

contoh : excel, lotus, dll

6. WIMP

- Windows Icon Menu Pointer

- Windows Icon Mouse Pulldown Menu

yang termasuk komponen WIMP : button, dialogue boxes, pallettes, dll

BIDANG STUDI/ILMU YANG BERPERAN

- Tujuan utama disusunnya berbagai cara interaksi manusia & komputer : untuk mempermudah manusia dalam mengoperasikan komputer dan mendapatkan berbagai umpan balik yang ia perlukan selama ia bekerja pada sebuah sistem komputer.

- Para perancang antarmuka manusia dan komputer berharap agar sistem komputer yang dirancangnya dapat bersifat akrab dan ramah dengan penggunanya (user friendly).
- Untuk membuat antarmuka yang baik dibutuhkan pemahaman beberapa bidang ilmu, antara lain :
 - Teknik elektronika & ilmu komputer = memberikan kerangka kerja untuk dapat merancang sistem HCI
 - Psikologi = memahami sifat & kebiasaan, persepsi & pengolahan kognitif, ketrampilan motorik pengguna
 - Perancangan grafis dan tipografi = sebuah gambar dapat bermakna sama dengan seribu kata. Gambar dapat digunakan sebagai sarana dialog cukup efektif antara manusia & komputer
 - Ergonomik = berhubungan dengan aspek fisik untuk mendapatkan lingkungan kerja yang nyaman, misal : bentuk meja & kursi kerja, layar tampilan, bentuk keyboard, posisi duduk, pengaturan lampu, kebersihan tempat kerja
 - Antropologi = ilmu pengetahuan tentang manusia, memberi suatu pandangan tentang cara kerja berkelompok yang masing – masing anggotanya dapat memberikan kontribusi sesuai dengan bidangnya
 - Linguistik = merupakan cabang ilmu yang mempelajari tentang bahasa. Untuk melakukan dialog diperlukan sarana komunikasi yang memadai berupa suatu bahasa khusus, misal bahasa grafis, bahasa alami, bahasa menu, bahasa perintah
 - Sosiologi = studi tentang pengaruh sistem manusia-komputer dalam struktur sosial, misal adanya PHK karena adanya otomasi kantor.

PRINSIP UTAMA MENDESAIN ANTARMUKA (INTERFACE)

Berikut ini beberapa hal yang menjadi prinsip utama mendesain antarmuka yang baik dengan memperhatikan karakteristik manusia & komputer :

1. User compatibility

- ◆ Antarmuka merupakan topeng dari sebuah sistem atau sebuah pintu gerbang masuk ke sistem dengan diwujudkan ke dalam sebuah aplikasi software.

- ◆ Oleh karena itu sebuah software seolah-olah mengenal user-nya, mengenal karakteristik user-nya, dari sifat sampai kebiasaan manusia secara umum.
- ◆ Desainer harus mencari dan mengumpulkan berbagai karakteristik serta sifat dari user karena antarmuka harus disesuaikan dengan user yang jumlahnya bisa jadi lebih dari 1 dan mempunyai karakter yang berbeda.
- ◆ Hal tersebut harus terpikirkan oleh desainer dan tidak dianjurkan merancang antarmuka dengan didasarkan pada dirinya sendiri
- ◆ Survey adalah hal yang paling tepat

2. Product compatibility

- ◆ Sebuah aplikasi yang bertopengkan antarmuka harus sesuai dengan sistem aslinya.
- ◆ Seringkali sebuah aplikasi menghasilkan hasil yang berbeda dengan sistem manual atau sistem yang ada.
- ◆ Hal tersebut sangat tidak diharapkan dari perusahaan karena dengan adanya aplikasi software diharapkan dapat menjaga produk yang dihasilkan dan menghasilkan produk yang jauh lebih baik.
- ◆ Contoh : aplikasi sistem melalui antarmuka diharapkan menghasilkan report/laporan serta informasi yang detail dan akurat dibandingkan dengan sistem manual.

3. Task compatibility

- ◆ Sebuah aplikasi yang bertopengkan antarmuka harus mampu membantu para user dalam menyelesaikan tugasnya. Semua pekerjaan serta tugas-tugas user harus diadopsi di dalam aplikasi tersebut melalui antarmuka.
- ◆ Sebisa mungkin user tidak dihadapkan dengan kondisi memilih dan berpikir, tapi user dihadapkan dengan pilihan yang mudah dan proses berpikir dari tugas-tugas user dipindahkan dalam aplikasi melalui antarmuka.
- ◆ Contoh : User hanya klik setup, tekan tombol next, next, next, finish, ok untuk menginstal suatu software.

4. Work flow compatibility

- ◆ Sebuah aplikasi sistem sudah pasti mengadopsi sistem manualnya dan didalamnya tentunya terdapat urutan kerja dalam menyelesaikan pekerjaan.
- ◆ Dalam sebuah aplikasi, software engineer harus memikirkan berbagai runutan-rununtan pekerjaan yang ada pada sebuah sistem.
- ◆ Jangan sampai user mengalami kesulitan dalam menyelesaikan pekerjaannya karena user mengalami kebingungan ketika urutan pekerjaan yang ada pada sistem manual tidak ditemukan pada software yang dihadapinya.
- ◆ Selain itu user jangan dibingungkan dengan pilihan-pilihan menu yang terlalu banyak dan semestinya menu-menu merupakan urutan dari runutan pekerjaan.
- ◆ Sehingga dengan workflow compatibility dapat membantu seorang user dalam mempercepat pekerjaannya.

5. Consistency

- ◆ Sebuah sistem harus sesuai dengan sistem nyata serta sesuai dengan produk yang dihasilkan.
- ◆ Banyak perusahaan dalam menjalankan sistemnya menggunakan aplikasi sistem yang berbeda di setiap divisi dalam perusahaan tersebut. Ada pula yang menggunakan aplikasi yang sama di divisi yang berbeda → seringkali keseragaman dalam menjalankan sistem tidak diperhatikan
- ◆ Oleh karena itu software engineer harus memperhatikan hal-hal yang bersifat konsisten pada saat merancang aplikasi khususnya antarmuka, contoh : penerapan warna, struktur menu, font, format desain yang seragam pada antarmuka di berbagai bagian, sehingga user tidak mengalami kesulitan pada saat berpindah posisi pekerjaan atau berpindah lokasi dalam menyelesaikan pekerjaan.
- ◆ Hal itu didasarkan pada karakteristik manusia yang mempunyai pemikiran yang menggunakan analogi serta kemampuan manusia dalam hal memprediksi.
- ◆ Contoh : keseragaman tampilan toolbar pada Word, Excell, PowerPoint, Access hampir sama.

6. Familiarity

- ◆ Sifat manusia mudah mengingat dengan hal-hal yang sudah sering dilihatnya/didapatkannya. Secara singkat disebut dengan familiar.
- ◆ Antarmuka sebisa mungkin didesain sesuai dengan antarmuka pada umumnya, dari segi tata letak, model, dsb.
- ◆ Hal ini dapat membantu user cepat berinteraksi dengan sistem melalui antarmuka yang familiar bagi user.

7. Simplicity

- ◆ Kesederhanaan perlu diperhatikan pada saat membangun antarmuka.
- ◆ Tidak selamanya antarmuka yang memiliki menu banyak adalah antarmuka yang baik.
- ◆ Kesederhanaan disini lebih berarti sebagai hal yang ringkas dan tidak terlalu berbelit.
- ◆ User akan merasa jengah dan bosan jika pernyataan, pertanyaan dan menu bahkan informasi yang dihasilkan terlalu panjang dan berbelit.
- ◆ User lebih menyukai hal-hal yang bersifat sederhana tetapi mempunyai kekuatan/bobot.

8. Direct manipulation

- ◆ User berharap aplikasi yang dihadapinya mempunyai media atau tools yang dapat digunakan untuk melakukan perubahan pada antarmuka tersebut.
- ◆ User ingin sekali aplikasi yang dihadapannya bisa disesuaikan dengan kebutuhan, sifat dan karakteristik user tersebut. Selain itu, sifat dari user yang suka merubah atau mempunyai rasa bosan.
- ◆ Contoh : tampilan warna sesuai keinginan (misal pink) pada window bisa dirubah melalui desktop properties, tampilan skin winamp bisa dirubah, dll.

9. Control

- ◆ Prinsip control ini berkenaan dengan sifat user yang mempunyai tingkat konsentrasi yang berubah-ubah. Hal itu akan sangat mengganggu proses berjalannya sistem.

- ◆ Kejadian salah ketik atau salah entry merupakan hal yang biasa bagi seorang user. Akan tetapi hal itu akan dapat mengganggu sistem dan akan berakibat sangat fatal karena salah memasukkan data 1 digit/1 karakter saja informasi yang dihasilkan sangat dimungkinkan salah.
- ◆ Oleh karena itu software engineer haruslah merancang suatu kondisi yang mampu mengatasi dan menanggulangi hal-hal seperti itu.
- ◆ Contoh : “illegal command”, “can’t recognize input” sebagai portal jika terjadi kesalahan.

10. WYSIWYG

- ◆ WYSIWYG = what you see is what you get = apa yang didapat adalah apa yang dilihatnya.
- ◆ Contoh : apa yang tercetak di printer merupakan informasi yang terkumpul dari data-data yang terlihat di layar monitor pada saat mencari data.
- ◆ Hal ini juga perlu menjadi perhatian software engineer pada saat membangun antarmuka.
- ◆ Informasi yang dicari/diinginkan harus sesuai dengan usaha dari user pada saat mencari data dan juga harus sesuai dengan data yang ada pada aplikasi sistem (software).
- ◆ Jika sistem mempunyai informasi yang lebih dari yang diinginkan user, hendaknya dibuat pilihan (optional) sesuai dengan keinginan user. Bisa jadi yang berlebihan itu justru tidak diinginkan user.
- ◆ Yang mendasar disini adalah harus sesuai dengan kemauan dan pilihan dari user.

10. Flexibility

- ◆ Fleksibel merupakan bentuk dari dari solusi pada saat menyelesaikan masalah.
- ◆ Software engineer dapat membuat berbagai solusi penyelesaian untuk satu masalah.
- ◆ Sebagai contoh adanya menu, hotkey, atau model dialog yang lainnya.

11. Responsiveness

- ◆ Setelah memberikan inputan atau memasukkan data ke aplikasi system melalui antarmuka, sebaiknya sistem langsung memberi tanggapan/respon dari hasil data yang diinputkan.
- ◆ Selain teknologi komputer semakin maju sesuai dengan tuntutan kebutuhan manusia, software yang dibangun pun harus mempunyai reaksi tanggap yang cepat. Hal ini didasari pada sifat manusia yang semakin dinamis / tidak mau menunggu.

12. Invisible Technology

- ◆ Secara umum, user mempunyai keingintahuan sebuah kecanggihan dari aplikasi yang digunakannya. Untuk itu aplikasi yang dibuat hendaknya mempunyai kelebihan yang tersembunyi. Bisa saja kelebihan itu berhubungan dengan sistem yang melingkupinya atau bisa saja kecanggihan atau kelebihan itu tidak ada hubungannya.
- ◆ Contoh : sebuah aplikasi mempunyai voice recognize sebagai media inputan, pengolah kata yang dilengkapi dengan language translator.

13. Robustness

- ◆ Interaksi manusia dan komputer (pembangunan antarmuka) yang baik dapat berupa frase-frase menu atau error handling yang sopan.
- ◆ Kata yang digunakan harus dalam kondisi bersahabat sehingga nuansa user friendly akan dapat dirasakan oleh user selama menggunakan sistem .
- ◆ Contoh yang kurang baik : YOU FALSE !!!, BAD FILES !!!, FLOPPY ERROR, dsb. Akan lebih baik jika BAD COMMAND OR FILES NAMES, DISK DRIVE NOT READY,dll.

14. Protection

- ◆ Suasana nyaman perlu diciptakan oleh software engineer di antarmuka yang dibangunnya.
- ◆ Nyaman disini adalah suasana dimana user akan betah dan tidak menemui suasana kacau ketika user salah memasukkan data atau salah eksekusi.

- ◆ Seorang user akan tetap merasa nyaman ketika dia melakukan kesalahan, misal ketika user melakukan deleting atau menghapus files tanpa sengaja tidaklah menjadi kekacauan yang berarti karena misal ada recovery tools seperti undo, recycle bin, dll atau “are you sure....”
- ◆ Proteksi disini lebih menjaga kenyamanan user ketika menggunakan aplikasi sistem khususnya data-data berupa file.

15. Ease Of Learning And Ease Of Use

- ◆ Kemudahan dalam mengoperasikan software hanya dengan memandangi atau belajar beberapa jam saja.
- ◆ Kemudahan dalam memahami icon, menu-menu, alur data software, dsb.
- ◆ Sesudah mempelajari, user dengan mudah dan cepat menggunakan software tersebut. Jika sudah memahami tentunya akan membantu proses menjalankan sistem dengan cepat dan baik.

Secara garis besar, pengembangan antarmuka perlu memperhatikan beberapa hal sebagai berikut :

- 1) Pengetahuan tentang mekanisme fungsi manusia sebagai pengguna komputer. Tentunya yang ada hubungannya dengan psikologi kognitif, tingkat perseptual, serta kemampuan motorik pengguna.
- 2) Berbagai informasi yang berhubungan berbagai informasi yang berhubungan dengan karakteristik dialog yang cukup lebar, seperti ragam dialog, struktur, isi tekstual dan grafis, waktu tanggap, dan kecepatan tampilan.
- 3) Penggunaan prototipe yang didasarkan pada spesifikasi dialog formal yang disusun secara bersama antara calon pengguna (user) dan perancang sistem, serta peranti bantu yang dapat digunakan untuk mempercepat proses pembuatan prototipe.
- 4) Teknik evaluasi yang digunakan untuk mengevaluasi hasil proses prototipe yang telah dilakukan, yaitu secara analitis berdasarkan pada analisis atas transaksi dialog, secara empiris menggunakan uji coba pada sejumlah kasus, umpan balik pengguna yang dapat dikerjakan dengan tanya jawab maupun kuesioner dan beberapa analisis yang dikerjakan oleh ahli antarmuka.

Kesulitan yang timbul dalam pengembangan fasilitas antarmuka dari sebuah perangkat lunak antara lain adalah :

- ◆ Antarmuka harus menangani beberapa piranti kontrol seperti adanya keyboard dan mouse maupun peripheral lainnya, yang semuanya mempunyai aliran data yang berbeda-beda dan mempunyai karakteristik yang berbeda pula.
- ◆ Waktu yang dibutuhkan pada saat pengiriman data. Bagaimana meyakinkan bahwa tidak terjadi keterlambatan antara tindakan dari pengguna dan respon/tanggapan dari sistem.

Untuk mempercepat proses perancangan dan pengembangan antarmuka, beberapa piranti bantu pengembang sistem antarmuka sering dimanfaatkan, seperti adanya perkembangan teknologi komputer Apple yang berfokus pada desain grafis, perkembangan teknologi pemrograman seperti Visual C/C++, Visual Basic, Delphi, Visual Foxpro, dll.

Dengan perkembangan itu kita dapat mendesain antarmuka yang luwes dan enak dipandang, bahkan cukup nyaman untuk digunakan dalam membuat topeng sebuah sistem.